


# Nginx Root Location Misconfiguration

## Nginx - Root Location Misconfiguration

15 Points 

Missing Root Location

Auteur

.Yo0x, 27 septembre 2024

Niveau ?



Validations

891 Challengeurs 1%

## Énoncé

Notre développeur web affirme que l'intranet qu'il a développé est sécurisé parce qu'il contient très peu de fonctionnalités. Prouvez-lui le contraire en consultant la configuration du serveur.

```
1. [server](http://wiki.nginx.org/NginxHttpCoreModule#server) {
2.     [listen](http://wiki.nginx.org/NginxHttpCoreModule#listen)      80;
3.     [server_name](http://wiki.nginx.org/NginxHttpCoreModule#server_name) _;
4.     [root](http://wiki.nginx.org/NginxHttpCoreModule#root) /etc/nginx;

5.     [location](http://wiki.nginx.org/NginxHttpCoreModule#location) = / {
6.         [return](http://wiki.nginx.org/NginxHttpRewriteModule#return) 302
        /login/login.html;
7.     }

8.     [location](http://wiki.nginx.org/NginxHttpCoreModule#location) /login/ {
9.         [alias](http://wiki.nginx.org/NginxHttpCoreModule#alias)
        /usr/share/nginx/html/login/;
10.    }

11.    [location](http://wiki.nginx.org/NginxHttpCoreModule#location) /static/ {
12.        [alias](http://wiki.nginx.org/NginxHttpCoreModule#alias)
        /var/www/app/static/;
13.    }

14.    [location](http://wiki.nginx.org/NginxHttpCoreModule#location) / {
15.        [try_files](http://wiki.nginx.org/NginxHttpCoreModule#try_files) $uri
        $uri/ =404;
16.        [default_type](http://wiki.nginx.org/NginxHttpCoreModule#default_type)
```

```
text/plain;

17.     }

18.     [error_page](http://wiki.nginx.org/NginxHttpCoreModule#error_page) 404 =200
/error.txt;

19.     [location](http://wiki.nginx.org/NginxHttpCoreModule#location) /error.txt {
20.         [internal](http://wiki.nginx.org/NginxHttpCoreModule#internal);
21.     }
22. }
```

## Faille

```
location / {
    try_files $uri $uri/ =404;
    default_type text/plain;
}
```

Avec :

```
root /etc/nginx;
```

Lorsque la requête n'est pas capturée par les `location /login/` ou `/static/`, elle est traitée par `location /` avec `root /etc/nginx`.

Cela signifie que **toute requête non capturée** sera résolue dans le dossier `/etc/nginx`.

### Correction :

- Utiliser un `root` pointant vers un dossier public (ex. `/var/www/html`).
- Ajouter des restrictions (`deny all;`, `internal`) sur les chemins sensibles.
- Éviter les `location /` trop génériques avec `try_files` mal sécurisé.

## Exploitation

```
server {
    listen      59093;
    server_name _;
    root /etc/nginx;

    location = / {
        return 302 /login/login.html;
    }

    location /login/ {
        alias /usr/share/nginx/html/login/;
    }

    location /static/ {
        alias /var/www/app/static/;
    }

    location / {
        try_files $uri $uri/ =404;
        default_type text/plain;
    }

    error_page 404 =200 /error.txt;

    location /error.txt {
        internal;
    }
}
```

Congratulation the flag is RM{b3\_C4r3fU1\_ab0uT\_R00t\_L0cation<3}